

CLAIMS

What is claimed is:

1. A method for controlling visibility of data during transaction processing in a
5 multi-version database management system, comprising:
 - receiving a request for a record from a requesting transaction, the requesting transaction having an associated transaction identifier which uniquely identifies the transaction, an invisibility list which identifies other transactions whose effects are to be invisible to the requesting transaction, and an isolation level which describes whether changes made by other transactions are to be visible to the transaction; and
 - determining whether the record is visible to the requesting transaction based on the isolation level of the requesting transaction, the transaction identifier, the invisibility list of the requesting transaction, and a creator transaction identifier in the requested record which identifies a transaction that created the record.
2. The method of claim 1 further comprising:
 - assigning a Record ID value to the record when the record is first created, the Record ID uniquely distinguishing the record from all other records, and the Record ID preserved across modifications of the record.
3. The method of claim 1 wherein the transaction identifier is a numeric value and transaction identifiers are assigned to transactions in increasing numerical order
25 based on the start time of the transaction, such that a first transaction can be determined to start before a second transaction if the transaction identifier associated with the first transaction is numerically less than the transaction identifier associated with the second transaction.

4. The method of claim 3 wherein transaction identifiers associated with transactions that operate in the present have an even numeric value, and transaction identifiers associated with transactions operating “as-of” a determined time in the past have an odd numeric value.
5
5. The method of claim 4 further comprising:
 - finding a transaction identifier for an earliest transaction that started on or after the specified “as-of” time;
 - 10 creating a new transaction, the new transaction having a start-time equal to the specified “as-of” time; a transaction identifier equal to the transaction identifier for the earliest transaction, less one; and an isolation level set to Read Committed; and
 - 15 initializing the invisibility list of the new transaction to include the transaction identifiers of all transactions having transaction identifiers values less than the transaction identifier for the new transaction and end-times greater than the specified “as-of” time.
6. The method of claim 5 further comprising:
 - 20 removing from the invisibility list of the new transaction, any transactions serialized before transactions visible to the new transaction; and
 - adding to a visibility list of the new transaction, any transactions with transaction identifiers greater than the transaction identifier of the new transaction, that are serialized before transactions visible to the new transaction.
- 25
7. The method of claim 1 wherein the requesting transaction further comprises an associated visibility list, the visibility list including numeric transaction identifiers which identify other transactions whose effects are to be visible to the requesting transaction.

8. The method of claim 1 additionally comprising:
 - initializing an invisibility list of the new transaction by:
 - searching a list of existing transactions to find transactions whose transaction identifier is less than the identifier of the new transaction, and whose state is an active state; and
 - inserting the transaction identifiers of such existing transactions into the invisibility list of the new transaction.
- 10 9. The method of claim 1 further comprising:
 - allowing a transaction to delete a record, by storing the transaction identifier of the transaction in a deleter transaction identifier field of the record.
10. The method of claim 1 further comprising:
 - allowing a transaction to delete a record, by adding a deletion descriptor to a list of deleted records, the deletion descriptor including the transaction identifier of the transaction performing the deletion and information uniquely identifying the record.
- 20 11. The method of claim 1 further comprising:
 - rolling back changes of a transaction, comprising:
 - examining all records that could have been created, updated or deleted by the transaction;
 - storing an aborted transaction identifier in a deleter transaction identifier field, if present, of records created by the requesting transaction, the aborted transaction identifier is less than a numeric value of any other non-NUL transaction identifier;

- storing a deletion descriptor in a list of deleted records, if present, the deletion descriptor including the aborted transaction identifier and information uniquely identifying the version of the record;
- 5 storing a NULL transaction identifier in the deleter transaction identifier field, if present, of records deleted by the requesting transaction;
- removing the deletion descriptor from a list of deleted records, if present, the deletion descriptor including the identifier of the requesting transaction; and
- 10 writing records with a modified deleter transaction identifier field to a persistent storage device.
12. The method of claim 11 wherein the examining step comprises:
- 15 starting the examination at a low-water-mark record and ending the examination at a high-water-mark record for a sequential set of records, the low-water-mark identifying a first record in the sequential set of records created, updated or deleted by the requesting transaction, and the high-water-mark identifying a last record in the sequential set of records created, updated or deleted by the requesting transaction.
- 20
13. The method of claim 1 further comprising:
- comprising:
- 25 determining a set of transactions, if any, that had started, but had neither committed nor aborted at the time the database had previously stopped operating;
- rolling back the changes for each such incomplete transaction;
- and

including the transaction identifiers of each such incomplete transaction on the invisibility lists of all new transactions started before the incomplete transactions have been rolled back.

- 5 14. The method of claim 2, further comprising:
- creating a new record by:
- obtaining a unique Record ID for the new record;
- storing the transaction identifier of the transaction creating the new record and the Record ID in the new record; and
- 10 storing a NULL transaction identifier value in a deleter transaction identifier field, if present, of the new record.
15. The method of claim 1 further providing the ability for a transaction to modify an existing record, comprising:
- creating a new record;
- copying fields from the existing record to the new record;
- storing a transaction identifier identifying the transaction in a creator-transaction-identifier field of the new record;
- copying updated data field values in the existing record into corresponding fields in the new record;
- 20 storing the transaction identifier of the transaction in a deleter transaction identifier field of the existing record, if present;
- adding a deletion descriptor to a list of deleted records, if the deleter transaction identifier field is not present, the deletion descriptor identifying the transaction and the existing record; and
- 25 storing a NULL transaction identifier value of the transaction performing the modification in a deleter transaction identifier field, if present, of the new record.

16. The method of claim 1 further comprising:
 - retrieving records visible to a transaction operating with Read Uncommitted isolation comprising:
 - reading a record from the database;
 - ensuring that the deleter transaction identifier of the record, if present, is a NULL transaction identifier; and
 - ensuring that a list of deleted records, if present, does not include a description of the record.
- 10 17. The method of claim 1 further comprising:
 - retrieving visible records for a requesting transaction operating with Serializable or Repeatable Read isolation comprising:
 - reading a record from the database; and
 - ensuring that the creator transaction identifier of the record is not on the invisibility list of the requesting transaction and has a value less than or equal to the transaction identifier of the requesting transaction; and
 - ensuring that the deleter transaction identifier of the record, if present, is a NULL transaction identifier, or is greater than the transaction identifier of the requesting transaction, or is both less than the transaction identifier of the requesting transaction and is stored on the invisibility list of the requesting transaction; and
 - ensuring that a list of deleted records, if present, does not include a description of the record, or that the record was deleted by a transaction whose identifier was greater than the transaction identifier of the requesting transaction, or that the record was deleted by a transaction whose identifier is both less than the transaction identifier of the requesting transaction and is stored on the invisibility list of the requesting transaction.
- 15
- 20
- 25

18. The method of claim 17 wherein records that are not visible to the requesting transaction, but that otherwise would meet the restrictions imposed by a database query, are flagged as being invisible and temporarily retained in memory.
19. The method of claim 7 further comprising:
 - removing a transaction identifier of a committing transaction from the invisibility list of a specified transaction operating with Read Committed isolation which may be the requesting transaction or some other transaction, if the transaction identifier of the committing transaction is less than the transaction identifier of the specified transaction; and
 - adding the transaction identifier of the committing transaction to the visibility list of the specified transaction, if the transaction identifier of the committing transaction is greater than the transaction identifier of the specified transaction.
20. The method of claim 19 further comprising:
 - retrieving visible records for the specified transaction operating with Read Committed isolation , comprising:
 - reading a record from the database; ensuring that either (a) the creator transaction identifier of the record is equal to the transaction identifier of the requesting transaction, or (b) the creator transaction identifier of the record is greater than the transaction identifier of the requesting transaction and the creator transaction identifier is on the visibility list of the requesting transaction or (c) the creator transaction identifier of the record is less than the transaction identifier of the

- requesting transaction and the creator transaction identifier of the record is not on the invisibility list of the requesting transaction;
- ensuring that the deleter transaction identifier of the record, if present, is either (a) a NULL transaction identifier; or (b) less than the transaction identifier of the requesting transaction and on the invisibility list of the requesting transaction or (c) greater than the transaction identifier of the requesting transaction and not on the visibility list of the requesting transaction; and
- ensuring that the list of deleted records, if present, does not include a description of the record, or that record was deleted by a transaction whose identifier was either (a) less than the transaction identifier of the requesting transaction and also on the invisibility list of the requesting transaction or (b) greater than the transaction identifier of the requesting transaction and not on the visibility list of the requesting transaction.
21. The method of claim 20 further comprising:
- avoiding unrepeatable reads comprising:
- tracking versions of records retrieved during the course of a transaction;
- checking if a different version of the same record is subsequently retrieved; and
- returning the version of the record originally retrieved or aborting the transaction.
22. The method of claim 1 wherein the step of determining whether the record is visible to the requesting transaction is carried out in a programmable filter.

23. The method of claim 22 wherein the programmable filter is implemented as a circuit component selected from the group consisting of Field Programmable Gate Array (FPGA), Application Specific Integrated Circuit (ASIC), Application Specific Standard Product (ASSP), discrete logic in a printed circuit board, and an programmable micro-processor.
5
24. The method of claim 1 further comprising:
retrieving a version of a record visible to a requesting transaction without reference to other versions of the record.
10
25. A multi-version database management system which controls visibility of data during transaction processing comprising:
15 a requesting transaction comprising:
an associated transaction identifier which uniquely identifies the requesting transaction;
an invisibility list which identifies other transactions whose effects are to be invisible to the requesting transaction; and
20 an isolation level which describes whether changes made by other transactions are to be visible to the requesting transaction; and
a transaction manager which receives a request for a record from the requesting transaction and determines whether the record is visible to the requesting transaction based on the isolation level of the requesting transaction, the transaction identifier, the invisibility list of the requesting transaction, and a creator transaction identifier in the requested record which identifies a
25 transaction that created the record.
26. The system of claim 25 further comprising:

a Record ID manager which assigns a Record ID value to the record when the record is first created, the Record ID uniquely distinguishing the record from all other records, and the Record ID preserved across modifications of the record.

5

27. The system of claim 25 wherein the transaction identifier is a numeric value and transaction identifiers are assigned to transactions in increasing numerical order based on the start time of the transaction, such that a first transaction can be determined to start before a second transaction if the transaction identifier associated with the first transaction is numerically less than the transaction identifier associated with the second transaction.
10
28. The system of claim 27 wherein transaction identifiers associated with transactions that operate in the present have an even numeric value, and transaction identifiers associated with transactions operating “as-of” a determined time in the past have an odd numeric value.
15
29. The system of claim 28 wherein the transaction manager finds a transaction identifier for an earliest transaction that started on or after the specified “as-of” time, creates a new transaction, the new transaction having a start-time equal to the specified “as-of” time; a transaction identifier equal to the transaction identifier for the earliest transaction, less one; and isolation level set to Read Committed; and initializes the invisibility list of the new transaction to include the transaction identifiers of all transactions having transaction identifiers values less than the transaction identifier for the new transaction and end-times greater than the specified “as-of” time.
20
25

30. The system of claim 29 wherein the transaction manager removes from the invisibility list of the new transaction, any transactions serialized before transactions visible to the new transaction and adds to a visibility list of the new transaction, any transactions with transaction identifiers greater than the transaction identifier of the new transaction, that are serialized before transactions visible to the new transaction.
5
31. The system of claim 25 wherein the requesting transaction further comprises:
an associated visibility list, the visibility list including numeric
10 transaction identifiers which identify other transactions whose effects are to be visible to the requesting transaction.
32. The system of claim 25 wherein to initialize an invisibility list of the new transaction, the transaction manager searches a list of existing transactions to find transactions whose transaction identifier is less than the identifier of the new transaction, and whose state is an active state and inserts the transaction identifiers of such existing transactions into the invisibility list of the new transaction.
15
- 20 33. The system of claim 25 wherein the transaction manager stores the transaction identifier of the transaction in a deleter transaction identifier field of the record to allow a transaction to delete a record.
- 25 34. The system of claim 25 wherein the transaction manager adds a deletion descriptor to a list of deleted records, the deletion descriptor including the transaction identifier of the transaction performing the deletion and information uniquely identifying the record to allow a transaction to delete a record.

35. The system of claim 25 wherein to roll back changes of a transaction the transaction manager
 - examines all records that could have been created, updated or deleted by the transaction,
 - 5 stores an aborted transaction identifier in a deleter transaction identifier field, if present, of records created by the requesting transaction, the aborted transaction identifier is less than a numeric value of any other non-NUL transaction identifier,
 - 10 stores a deletion descriptor in a list of deleted records, if present, the deletion descriptor including the aborted transaction identifier and information uniquely identifying the version of the record,
 - 15 stores a NULL transaction identifier in the deleter transaction identifier field, if present, of records deleted by the requesting transaction,
 - removes the deletion descriptor from a list of deleted records, if present, the deletion descriptor including the identifier of the requesting transaction, and
 - 15 writes records with a modified deleter transaction identifier field to a persistent storage device.
36. The system of claim 35 wherein to examine all records, the transaction manager starts the examination at a low-water-mark record and ending the examination at a high-water-mark record for a sequential set of records, the low-water-mark identifying a first record in the sequential set of records created, updated or deleted by the requesting transaction, and the high-water-mark identifying a last record in the sequential set of records created, updated or deleted by the requesting transaction.
- 25
37. The system of claim 25 wherein to perform online recovery support for transaction processing, the transaction manager

determines a set of transactions, if any, that had started, but had neither committed nor aborted at the time the database had previously stopped operating,

5 rolls back the changes for each such incomplete transaction and includes the transaction identifiers of each such incomplete transaction on the invisibility lists of all new transactions started before the incomplete transactions have been rolled back.

- 10 38. The system of claim 26 wherein to create a new record, a Query Execution manager obtains a unique Record ID for the new record from the RecordID manager, stores the transaction identifier of the transaction creating the new record and the Record ID in the new record and stores a NULL transaction identifier value in a deleter transaction identifier field, if present, of the new record.

15 39. The system of claim 25 wherein to modify an existing record, the transaction manager

20 requests a Query execution manager to create a new record,

 copies fields from the existing record to the new record,

 stores a transaction identifier identifying the transaction in a creator-transaction-identifier field of the new record,

 copies updated data field values in the existing record into corresponding fields in the new record,

 stores the transaction identifier of the transaction in a deleter transaction identifier field of the existing record, if present,

 adds a deletion descriptor to a list of deleted records, if the deleter transaction identifier field is not present, the deletion descriptor identifying the transaction and the existing record, and

stores a NULL transaction identifier value of the transaction performing the modification in a deleter transaction identifier field, if present, of the new record.

- 5 40. The system of claim 25 wherein to retrieve records visible to a transaction operating with Read Uncommitted isolation, the transaction manager reads a record from the database, ensures that the deleter transaction identifier of the record, if present, is a NULL transaction identifier and ensures that a list of deleted records, if present, does not include a description of the record.

10

41. The system of claim 25 wherein to retrieve visible records for a requesting transaction operating with Serializable or Repeatable Read isolation, the transaction manager
reads a record from the database and ensures that the creator transaction identifier of the record is not on the invisibility list of the requesting transaction and has a value less than or equal to the transaction identifier of the requesting transaction,

15

20

25

ensures that the deleter transaction identifier of the record, if present, is a NULL transaction identifier, or is greater than the transaction identifier of the requesting transaction, or is both less than the transaction identifier of the requesting transaction and is stored on the invisibility list of the requesting transaction and
ensures that a list of deleted records, if present, does not include a description of the record, or that the record was deleted by a transaction whose identifier was greater than the transaction identifier of the requesting transaction, or that the record was deleted by a transaction whose identifier is both less than the transaction identifier of the requesting transaction and is stored on the invisibility list of the requesting transaction.

42. The system of claim 41 wherein records that are not visible to the requesting transaction, but that otherwise would meet the restrictions imposed by a database query, are flagged as being invisible and temporarily retained in memory.
43. The system of claim 31 wherein the transaction manager removes a transaction identifier of a committing transaction from the invisibility list of a specified transaction operating with Read Committed isolation which may be the requesting transaction or some other transaction, if the transaction identifier of the committing transaction is less than the transaction identifier of the specified transaction; and adds the transaction identifier of the committing transaction to the visibility list of the specified transaction, if the transaction identifier of the committing transaction is greater than the transaction identifier of the specified transaction.
44. The system of claim 43 wherein to retrieve visible records for the specified transaction operating with Read Committed isolation, the transaction manager reads a record from the database; ensuring that either (a) the creator transaction identifier of the record is equal to the transaction identifier of the requesting transaction, or (b) the creator transaction identifier of the record is greater than the transaction identifier of the requesting transaction and the creator transaction identifier is on the visibility list of the requesting transaction or (c) the creator transaction identifier of the record is less than the transaction identifier of the requesting transaction and the creator transaction identifier of the record is not on the invisibility list of the requesting transaction;

- ensures that the deleter transaction identifier of the record, if present, is either (a) a NULL transaction identifier; or (b) less than the transaction identifier of the requesting transaction and on the invisibility list of the requesting transaction or (c) greater than the transaction identifier of the requesting transaction and not on the visibility list of the requesting transaction; and
- ensures that the list of deleted records, if present, does not include a description of the record, or that record was deleted by a transaction whose identifier was either (a) less than the transaction identifier of the requesting transaction and also on the invisibility list of the requesting transaction or (b) greater than the transaction identifier of the requesting transaction and not on the visibility list of the requesting transaction.
45. The system of claim 44 wherein to avoid unrepeatable reads, the transaction manager tracks versions of records retrieved during the course of a transaction, checks if a different version of the same record is subsequently retrieved and returns the version of the record originally retrieved or aborting the transaction.
46. The system of claim 25 further comprising:
a programmable filter which determines whether the record is visible to
the requesting transaction.
47. The system of claim 46 wherein the programmable filter is implemented as a circuit component selected from the group consisting of Field Programmable Gate Array (FPGA), Application Specific Integrated Circuit (ASIC), Application Specific Standard Product (ASSP), discrete logic in a printed circuit board, and an programmable micro-processor.

48. The system of claim 25 wherein the transaction manager retrieves a version of a record visible to a requesting transaction without reference to other versions of the record.
- 5 49. A multi-version database management system which controls visibility of data during transaction processing, comprising:
- means for receiving a request for a record from a requesting transaction, the requesting transaction having an associated transaction identifier which uniquely identifies the transaction, an invisibility list which identifies other transactions whose effects are to be invisible to the requesting transaction, and an isolation level which describes whether changes made by other transactions are to be visible to the transaction; and
- means for determining whether the record is visible to the requesting transaction based on the isolation level of the requesting transaction, the transaction identifier, the invisibility list of the requesting transaction, and a creator transaction identifier in the requested record which identifies a transaction that created the record.